**Q.3:**In this same GitHub repository, create a new branch ‘geometry-calculator’, we'll work on a simple Python program that calculates the area of a circle and the area of a rectangle. We'll use Git stash to switch between working on multiple features (calculating circle area and calculating rectangle area) without committing incomplete changes.

*import math*

*class GeometryCalculator:*

*def calculate\_circle\_area(self, radius):*

*return math.pi \* radius \*\* 2*

*def calculate\_rectangle\_area(self, length, width):*

*return length \* width*

*if \_\_name\_\_ == "\_\_main\_\_":*

*calculator = GeometryCalculator()*

*# TODO: Implement the feature to calculate the area of a circle*

*# radius = 5*

*# print(f"The area of the circle with radius {radius} =*

*{calculator.calculate\_circle\_area(radius)}")*

*# TODO: Implement the feature to calculate the area of a rectangle # length = 10*

*# width = 6*

*# print(f"The area of the rectangle with length {length} and width {width} = {calculator.calculate\_rectangle\_area(length, width)}")*

Workflow Steps:

a. Create a New Branch:

- Create a new branch named "feature/circle-area" to work on the circle area feature

b. Stash Changes for Circle Area Feature:

- Before committing the changes, stash them using *git stash*to save the incomplete feature implementation.

- Verify that the working directory is clean

c. Create a New Branch for Rectangle Area Feature:

- Create a new branch named "feature/rectangle-area" to work on the rectangle area

d. Stash Changes for Rectangle Area Feature:

- Before committing the changes, stash them using *git stash*to save the incomplete feature implementation.

- Verify that the working directory is clean

e. Switch Back to Circle Area Branch:

- Switch back to the "feature/circle-area" branch to continue working on the circle area feature.

- Retrieve the stashed changes

- Complete the circle area feature implementation and save the changes.

f. Commit and Push Circle Area Feature:

g. Switch Back to Rectangle Area Branch:

- Switch back to the "feature/rectangle-area" branch to continue working on the rectangle area feature.

- Retrieve the stashed changes

- Complete the rectangle area feature implementation and save the changes.

h. Commit and Push Rectangle Area Feature

i. Create Pull Requests:

- Create a pull request to the ‘dev’ branch.

j. Review and Merge

- Have another team member or reviewer review your pull requests. - After receiving approval, merge both pull requests into the main branch.

Documentation:

**Git Workflow Documentation**

**Project: Geometry Calculator**

**Objective**

To develop a Python program that calculates the area of a circle and a rectangle while using Git best practices, including branching, stashing, pull requests (PRs), and reviews.

**Workflow Steps**

**1. Cloning the Repository**

git clone https://github.com/YourUsername/git\_assignment\_HeroVired.git

cd git\_assignment\_HeroVired

**2. Creating Feature Branches**

**Create a New Branch for the Circle Area Feature**

git checkout -b feature/circle-area

**Stash Uncommitted Changes**

git stash

**Create a New Branch for the Rectangle Area Feature**

git checkout -b feature/rectangle-area

**Stash Uncommitted Changes**

git stash

**3. Implementing and Committing Changes**

**Switch Back to Circle Area Branch & Retrieve Stashed Changes**

git checkout feature/circle-area

git stash pop

* Implement the circle area calculation feature.

**Commit and Push**

git add geometry\_calculator.py

git commit -m "Added circle area calculation"

git push origin feature/circle-area

**Switch Back to Rectangle Area Branch & Retrieve Stashed Changes**

git checkout feature/rectangle-area

git stash pop

* Implement the rectangle area calculation feature.

**Commit and Push**

git add geometry\_calculator.py

git commit -m "Added rectangle area calculation"

git push origin feature/rectangle-area

**4. Creating Pull Requests (PRs)**

 







